CLAIMS

1. A method of verifying test data for testing an integrated circuit device having multiple device time domains each having a cycle duration, comprising:

selecting a virtual tester time domain having a cycle duration,

if the cycle duration of the virtual tester time domain is equal to the cycle duration of one of the multiple device time domains, translating the test data for each device time domain other than said one time domain to the virtual tester time domain and otherwise translating the test data for each device time domain to the virtual tester time domain, and

applying the translated test data to a device logic simulator that simulates the integrated circuit device.

2. A method according to claim 1, comprising, for each device time domain:

determining whether the time domain is complex or simple,

in the event that all time domains are simple, selecting the time domain with the longest cycle duration as the virtual tester time domain,

in the event that only one time domain is complex, selecting the complex time domain as the virtual tester time domain,

in the event that at least two time domains are complex, selecting the complex time domain having the shortest cycle duration as the virtual tester time domain.

3. A method according to claim 1, comprising:

analyzing the test data and identifying portions of the test data corresponding to the device time domains respectively and, for each device time domain, determining whether the time domain is complex or simple based on the test data,

in the event that all device time domains are simple, selecting the time domain with the longest cycle duration as the virtual tester time domain,

in the event that only one device time domain is complex, selecting the complex time domain as the virtual tester time domain,

in the event that at least two device time domains are complex, selecting the complex time domain having the shortest cycle duration as the virtual tester time domain.

4. A method according to claim 1, comprising applying the test data to the device logic simulator using a virtual tester and a virtual tester interface that operate synchronously with the device logic simulator.

5. A virtual tester for evaluating primary test data for testing a physical device that implements a device logic simulator and has multiple time domains, by transforming the primary test data to a format that is compatible with a virtual tester interface and applying the transformed test data through the virtual tester interface to the device logic simulator, the virtual tester comprising:

a first means for examining the primary test data and determining whether at least one of the time domains is complex and, in the event that only one time domain is complex, selecting the complex time domain, and in the event that at least two time domains are complex, selecting the complex time domain that has the shortest cycle duration,

a second means for receiving the primary test data for a physical tester domain other than the selected complex domain, mapping time of occurrence of an event in said physical tester domain to a corresponding time in the selected complex domain, and generating secondary test data for said physical tester domain wherein the time of occurrence of said event is specified by reference to a cycle boundary of the selected complex time domain, and

a third means responsive to a master clock signal at a predetermined frequency for transforming the secondary test data into a format that is compatible with the synchronous virtual tester interface.

6. A virtual tester according to claim 5, wherein the first means selects the time domain that has the longest cycle duration in the event that the first means determines that no time domain is complex.

7. Apparatus for evaluating primary test data for testing a physical device having multiple time domains, the apparatus comprising a virtual tester and a virtual tester interface, the virtual tester including:

a first means for examining the primary test data and determining whether at least one of the time domains is complex and, in the event that only one time domain is complex, selecting the complex time domain, and in the event that at least two time domains are complex, selecting the complex time domain that has the shortest cycle duration,

a second means for receiving the primary test data for a physical tester domain other than the selected complex domain, mapping time of occurrence of an event in said physical tester domain to a corresponding time in the selected complex domain, and generating secondary test data for said physical tester domain wherein the time of occurrence of said event is specified by reference to a cycle boundary of the selected complex time domain,

a third means responsive to a master clock signal at a predetermined frequency for transforming the secondary test data into a format that is compatible with the virtual tester interface,

and the virtual tester interface being located functionally between the virtual tester and the device logic simulator, the virtual tester interface being responsive to said master clock signal for passing signals between the virtual tester and the device logic simulator.

8. A method of carrying out virtual-cycle-based and digital-pin-based communication of test events between a virtual tester and a device logic simulator in a test plan that includes force and compare events, comprising:

determining, over all virtual cycles and all digital pins, the highest number of force events at a digital pin in a virtual cycle,

determining, over all virtual cycles and all digital pins, the highest number of compare events at a digital pin in a virtual cycle,

creating a data structure having, for each virtual cycle and each digital pin, entries for said highest number of force events and said highest number of compare events,

for a given virtual cycle and a given digital pin for which the number of force events is less than said highest number of force events, populating the data structure with at least one value that defines a valid force event at the given digital pin in the given virtual cycle and at least one value that does not define a

valid force event at the given digital pin in the given virtual cycle,

for a given virtual cycle and a given digital pin for which the number of compare events is less than said highest number of compare events, populating the data structure with at least one value that defines a valid compare event at the given digital pin in the given virtual cycle and at least one value that does not define a valid compare event at the given digital pin in the given virtual cycle.

9. A method according to claim 8, further comprising, for a given virtual cycle and a given digital pin for which the number of compare events is zero and the number of force events is zero, populating the data structure with values that do not define either a valid force event or a valid compare event at the given digital pin in the given virtual cycle.

10. A computer readable medium containing software which, when read and executed by a computer, causes the computer to carry out a method of verifying test data for testing an integrated circuit device having multiple device time domains each having a cycle duration, comprising:

selecting a virtual tester time domain having a cycle duration,

if the cycle duration of the virtual tester time domain is equal to the cycle duration of one of the multiple device time domains, translating the test data for each device time domain other than said one time domain to the virtual tester time domain and otherwise translating the test data for each device time domain to the virtual device time domain,

applying the translated test data to the a device logic simulator that simulates the integrated circuit device.

11. A computer readable medium containing software which, when read and executed by a computer, causes the computer to carry out a method of of carrying out virtual-cycle-based and digital-pin-based communication of test events between a virtual tester and a device logic simulator in a test plan that includes force and compare events, comprising:

determining, over all virtual cycles and all digital pins, the highest number of force events at a digital pin in a virtual cycle,

determining, over all virtual cycles and all digital pins, the highest number of compare events at a digital pin in a virtual cycle,

creating a data structure having, for each virtual cycle and each digital pin, entries for said highest number of force events and said highest number of compare events,

for a given virtual cycle and a given digital pin for which the number of force events is less than said highest number of force events, populating the data structure with at least one value that defines a valid force event at the given digital pin in the given virtual cycle and at least one value that does not define a valid force event at the given digital pin in the given virtual cycle,

for a given virtual cycle and a given digital pin for which the number of compare events is less than said highest number of compare events, populating the data structure with at least one value that defines a valid compare event at the given digital pin in the given virtual cycle and at least one value that does not define a valid compare event at the given digital pin in the given virtual cycle.

12. A computer readable medium according to claim 11, wherein the software, when read and executed by the computer, causes the computer to carry out a method that further comprises, for a given virtual cycle and a given digital pin for which the number of compare events is zero and the number of force events is zero, populating the data structure with values that do not define either a valid force event or a valid compare event at the given digital pin in the given virtual cycle.